

# Rook Jumping Maze

---

Starting at the circled square in the upper-left corner, find a path to the goal square marked "G". From each numbered square, one may move that exact number of squares horizontally or vertically in a straight line.

				
				
				
				
				

The solution to this maze is 13 jumps.

[More information on Rook Jumping Mazes](#) (including the solution to the maze above)

## PART I

**Question 1:** What algorithm would you use to solve a rook jumping maze. Why? If a heuristic is used, what is the heuristic?

**Question 2:** What is the branching factor of a state search algorithm that starts at the circle and ends at the G? What is the branching factor of a state search algorithm that starts at the G and ends at the circle? Should one implement forward search or backward search?

## **PART II**

Suppose instead of solving a rook jumping maze, we wanted to *generate* a rook jumping maze. That is, given a completely blank grid, what numbers should we put into each cell, and where should the G go? If we could do this, we could create a system that endlessly generates rook jumping mazes for other people to play.

Not all rook jumping mazes are created equally. For example, putting a 1 in every cell would make for a very easy maze. Choosing random numbers for each cell might result in an un-solvable maze. In this part, we look at how to use hill-climbing to generate solvable rook jumping mazes of differing difficulties.

**Question 3:** What is the state representation we should use for a hill-climbing algorithm? What should the start state look like?

**Question 4:** Describe a neighborhood and a successor function.

**Question 5:** Describe an evaluation function that converts a possible maze into a real number. The evaluation function we want to consider: (a) the solvability of the maze, (b) how many dead-ends (where there are no successors to a cell), (c) loops, (d) length of shortest solution, (e) number of solutions, (f) whether easier to solve going forward or backward.

**Question 6:** Which optimization search should one choose?